Operational Research Club

•••

First meeting

Why do we need Operational Research

1. Effective decisons
2. Better coordination
3. Facilitates control
4. Improves productivity

Operational Research Methods

Linear Programming

Non-linear Programming

Integer linear Programming

→ Maximization/ Minimaztion or in general to find **optimal outcome**

Elements you need

1. Objective functions [your demand]
2. Variables [what are involved]
3. Constraints [limitations]

AMPL

- A Mathematical Programming Language --

algebraic modeling language to describe and solve high-complexity problems for large-scale mathematical computing (i.e. Optimization)

AMPL solver tables

1. Linear (simplex & interior)
2. Network
3. Quardratic
4. Non-linear
5. Integer (linear & non-linear)

AMPL

Basics files:

1. .mod - used to declare the elements of the models: variables, objective, constraints and data (sets and parameters).
2. .dat - used to define the data for the model.
3. .run - where variable configurations are defined, “scripting constructs,” such as reading tables or data bases.

Sintaxe:

1. Variable: var VariableName;
2. Objective: minimize or maximize ObjectiveName: . . . ;
3. Constraint: subject to RestrictionName: . . . ;

Examples

Pen Deals produces two colors of pen, blue and black. Blue pen is sold for 1 euro per pen, while black pen is sold for 1.5 euro per pen. The company owns a process plant which can produce one color at a time. However, blue pen is produced at a rate of 40 pens per hour, while the production rate for black pen is 30 pens per hour. Besides, the marketing department estimates that at most 860 pens of black color and 1000 pens of blue color can be sold in the market. During a week, the plant can operate for 40 hours and the pens can be stored for the following week.

Determine how many pens of each pen should be produced to maximize week revenue.

Mathematical approach:

Variables: blue pen, black pen

Objective function: 1* # of sold blue pen + 1.5 * # of sold black pen

Constraints:

a. 1/40* # of blue pen + 1/30* # of black pen
b. blue pen <= 1000
c. black pen <= 860
d. blue pen, black pen >= 0

Remark

1. Every line instruction must be terminated with “;”.
2. Line comments are preceded by the symbol “#”.
3. Block commands are enclosed by the symbols “//*. . . *//”.
4. AMPL is “case-sensitive”. I Variable names must be unique.

Ampl approach

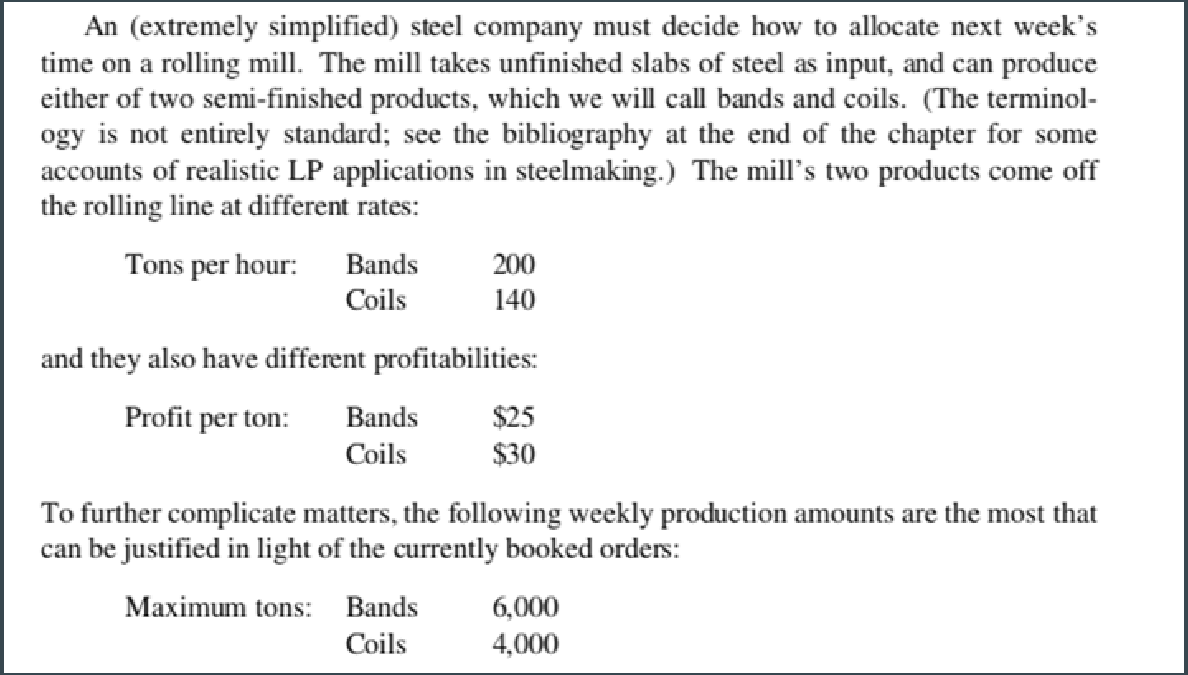
(file extension: .mod)

```
var BluePen;
var BlackPen;

maximize Revenue: 10*BluePen + 15*BlackPen;

subject to Time: (1/40)*BluePen + (1/30)*BlackPen <= 40;
subject to BlueLimit: 0 <= BluePen <= 1000;
subject to BlackLimit: 0 <= BlackPen <= 860;
```

Warm up



An (extremely simplified) steel company must decide how to allocate next week's time on a rolling mill. The mill takes unfinished slabs of steel as input, and can produce either of two semi-finished products, which we will call bands and coils. (The terminology is not entirely standard; see the bibliography at the end of the chapter for some accounts of realistic LP applications in steelmaking.) The mill's two products come off the rolling line at different rates:

Tons per hour:	Bands	200
	Coils	140

and they also have different profitabilities:

Profit per ton:	Bands	\$25
	Coils	\$30

To further complicate matters, the following weekly production amounts are the most that can be justified in light of the currently booked orders:

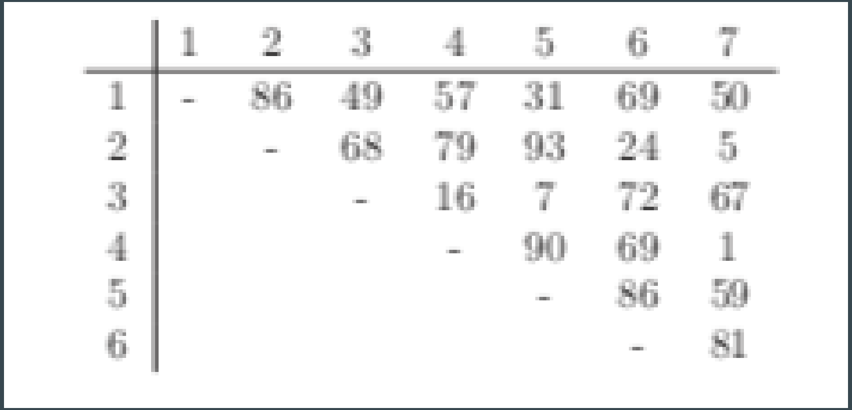
Maximum tons:	Bands	6,000
	Coils	4,000

Question:

If 40 hours of production time are available this week, how many tons of bands and how many tons of coils should be produced to bring in the greatest total profit?

Travelling Salesman problem

A travelling salesman must visit 7 customers in 7 different locations whose (symmetric) distance matrix is:

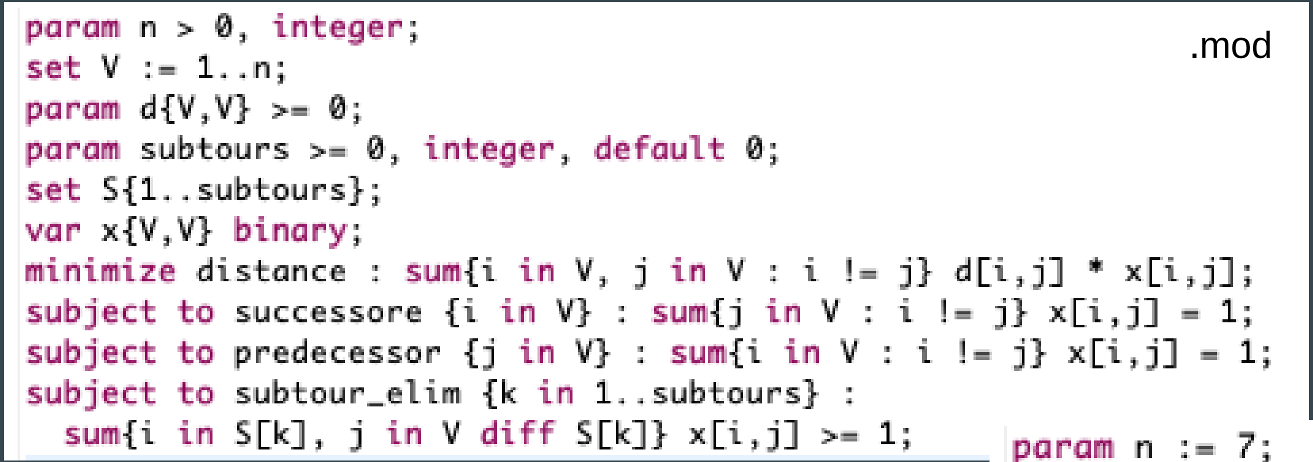


	1	2	3	4	5	6	7
1	-	86	49	57	31	69	50
2		-	68	79	93	24	5
3			-	16	7	72	67
4				-	90	69	1
5					-	86	59
6						-	81

Herustic method

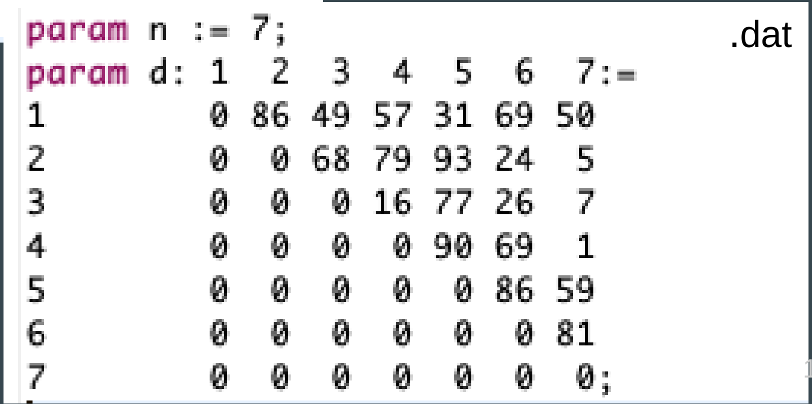
1. variable: Xij , indices of the customers, =1 the salesman has travelled; =0 otherwise
2. parameter: Dij, distance between i and j
3. objective function: min∑DijXij
4. constraints: i,j >=0

Ampl method



.mod

```
param n > 0, integer;
set V := 1..n;
param d{V,V} >= 0;
param subtours >= 0, integer, default 0;
set S{1..subtours};
var x{V,V} binary;
minimize distance : sum{i in V, j in V : i != j} d[i,j] * x[i,j];
subject to successore {i in V} : sum{j in V : i != j} x[i,j] = 1;
subject to predecessor {j in V} : sum{i in V : i != j} x[i,j] = 1;
subject to subtour_elim {k in 1..subtours} :
  sum{i in S[k], j in V diff S[k]} x[i,j] >= 1;
```



.dat

```
param n := 7;
param d: 1  2  3  4  5  6  7:=
1        0 86 49 57 31 69 50
2        0  0 68 79 93 24  5
3        0  0  0 16 77 26  7
4        0  0  0  0 90 69  1
5        0  0  0  0  0 86 59
6        0  0  0  0  0  0 81
7        0  0  0  0  0  0  0;
```

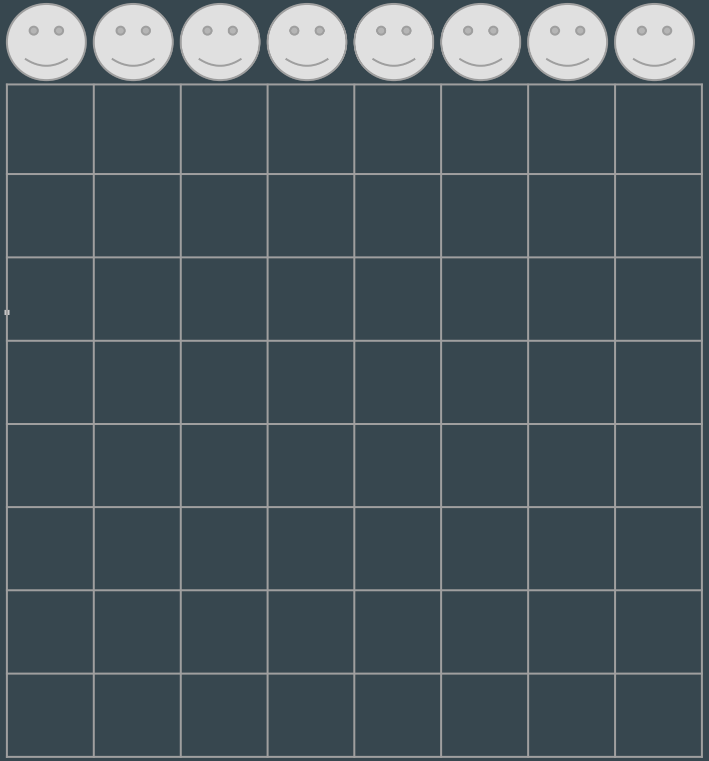
Integer programming

Eight queen question:

Display each queen commands **a row**,

a column and **two diagonals** on the chessboard.

How can we shorten the process time?



solution

1. set variables : x_{ij} where i = row, j = colomn;

   1 = there is a queen; 0 = otherwise

1. parameters : n, number of cells on one side of the (square) chessboard = 8

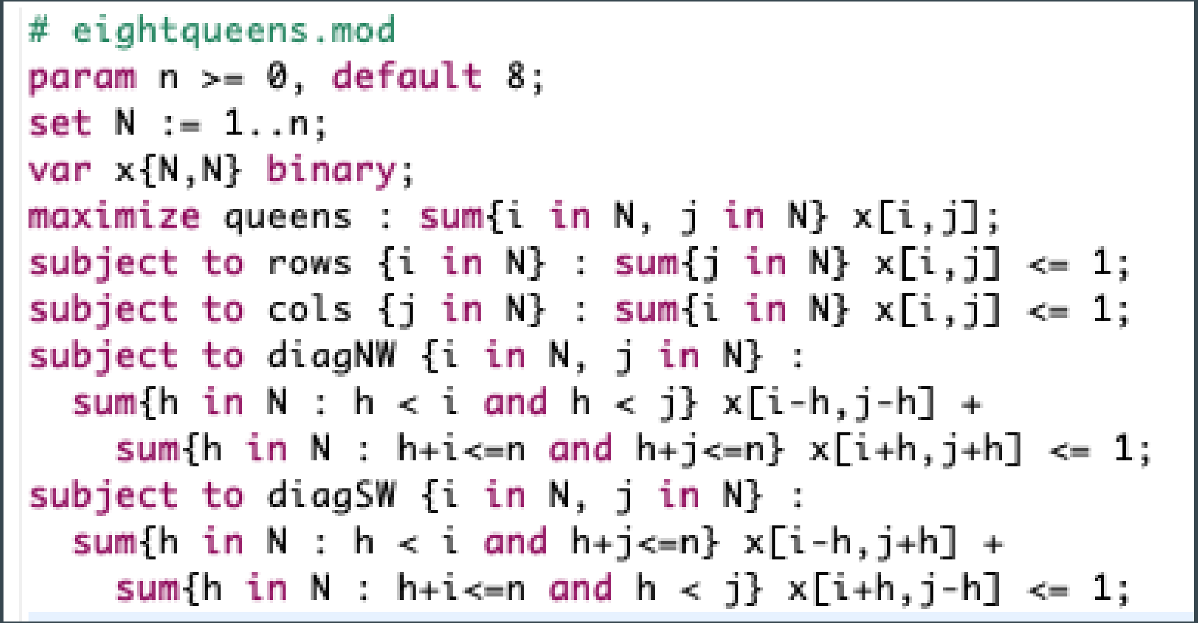
3. constraints : 1. $\forall i \le n(\sum x_{ij} \le 1)$; 2. $\forall j \le n(\sum x_{ij} \le 1)$;

   3. $\sum x_{ij} <= 8$;

with CPLEX solver

11	12	13	...				18
21	22						
31		33					
...			...				
81							88

Ampl method



```
# eightqueens.mod
param n >= 0, default 8;
set N := 1..n;
var x{N,N} binary;
maximize queens : sum{i in N, j in N} x[i,j];
subject to rows {i in N} : sum{j in N} x[i,j] <= 1;
subject to cols {j in N} : sum{i in N} x[i,j] <= 1;
subject to diagNW {i in N, j in N} :
  sum{h in N : h < i and h < j} x[i-h,j-h] +
    sum{h in N : h+i<=n and h+j<=n} x[i+h,j+h] <= 1;
subject to diagSW {i in N, j in N} :
  sum{h in N : h < i and h+j<=n} x[i-h,j+h] +
    sum{h in N : h+i<=n and h < j} x[i+h,j-h] <= 1;
```

Advanced questions

1. Telecomunication networking
2. Non-linear programming
3. Piecewise-linear program

Question?

What’s next:

more Maximiztion and Minimization with Ampl

Feedback???